

VBA: Hunting Duplicate Polygons in a Large Polygon Dataset

Contributed by Bert Granberg
06, May. 2009
Last Updated 06, May. 2009

It was recently brought to my attention that one of the SGID's large vegetation data layers (SWReGAP) had many duplicate polygons. Sure enough, in some locations spatial queries and identify operations would return duplicate features...same attributes, same geometry, same area, same perimeter, BUT two or more different object IDs.

These errors were likely introduced during the raster to vector conversion process after AGRC received the original data from its creators at Utah State.

As bad luck would have it, there was no discernable pattern to isolate the duplicates and the dataset had ONLY 3.6 million features. Since this is polygon dataset is supposed to be planar (no polygons should overlap with other polygons), I went ahead on the assumption that if two features had the same centroid, area, and perimeter, then one of them was a duplicate. Since the data was already in a geodatabase, I already had access to an area and perimeter (length) attribute values for each polygon. I just needed to add a text field to store the centroid coordinate, do a three tiered sort, and iterate through the sorted records to identify duplicates.

Not sure if this is the best approach but here is what I did to find these features.

- Add a field called LabelPt, (Text 100) and calculate its value to the concatenation of the X and Y coordinate resulting from the ArcObjects method IArea.LabelPoint which guarantees that the point returned (unlike IArea.Centroid) will be within the given polygon. My LabelPt values (in UTM coords) look like this --> "379891.217607 4182762.234845"
- Run a Visual Basic for Applications script that first sorts by LabelPt attribute value (from Step 1), then by area, then by perimeter. Then a looping structure looks through all the sorted features checking to see which features have the same LabelPt, area, and length attribute combination as the sorted feature before it. These are marked and added to a selected set.
- I had problems running out of memory so I run this script in batches of 1 million records and record the last LabelPt value after each batch of 1 million records. I stop the script after a million records have been processed and then right click on the layer and 'Create a Layer from Selected Records'. Then I open up the ArcGIS command line and use the DeleteFeatures command against the new selection layer.
- After the I then modify the filter on the TableSort object for the next run to look for everything greater than the LabelPt value for the previous million records. The corrected SWReGAP vegetation layer will be available in the new SGID 9.3 Database which will go online in a week or so.

Code:

Step #1 Calc Script for ArcGIS Field Calculator:

VBA Pre-Logic Expression Box:

```
Dim pArea as IArea
Dim pPoint as IPoint
Set pArea = [Shape]
Set pPoint = pArea.LabelPoint
```

Bottom Box:

```
ppoint.x & " " & ppoint.y
```

Step #2 VBA Script to Sort, Iterate, and Compare to Mark Duplicates

Public Sub checkforPolygonDuplicates()

```
'Get Reference To Current ArcMap Session
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

Dim pFLayer As IFeatureLayer
Dim pFClass As IFeatureClass
```

```
Dim pTable As ITable
Dim pQF As IQueryFilter
Set pQF = New QueryFilter
```

```
*****
```

```
'SET THIS: Get the first layer in the table of contents, layer 0
'the featureclass you are using must be referenced by the
'first layer in the map
Set pFLayer = pMap.Layer(0)
```

```
'SET THIS & UNCOMMENT ONLY IF you are doing multiple batches of processing
'set it to the maximum labelpt x coordinate from the last
'batch of processing
```

```
'pQF.WhereClause = "left(LabelPt,6) > '383893"
```

```
*****
```

```
Set pFClass = pFLayer.FeatureClass
Set pTable = pFClass ' QI
```

```
Dim labelPtFI As Long
Dim areaFI As Long
Dim perimeterFI As Long
```

```
labelPtFI = pTable.FindField("LabelPt")
areaFI = pTable.FindField("Shape.area")
perimeterFI = pTable.FindField("Shape.len")
```

```
'Get the selected set of polygon features
Dim pFSel As IFeatureSelection
Dim pSelSet As ISelectionSet
Set pFSel = pFLayer 'QI
Set pSelSet = pFSel.SelectionSet
```

```
'set up tablesort to return results of a multi-field sort
Dim pTableSort As ITableSort
Set pTableSort = New TableSort 'VBA Query Interface
'the fields used and their order or precedence for the sort
pTableSort.Fields = ("LabelPt,Shape.len,Shape.area")
pTableSort.Ascending("Shape.area") = True
pTableSort.Ascending("Shape.len") = True
pTableSort.Ascending("LabelPt") = True
Set pTableSort.QueryFilter = pQF
```

```
'sort on only selected features
Set pTableSort.Table = pTable
```

```
pTableSort.Sort Nothing
```

```
'get a cursor to iterate thru the features returned by the tablesort
Dim pCursor As ICursor
Dim pRow As IRow
Set pCursor = pTableSort.Rows
```

```
Dim xYAPStr As String 'concatenate xy coord of label point with area and perimeter
Dim lastXYAPStr As String 'track xyAPStr from previous records
Dim Count As Long
lastXYAPStr = ""
```

```
Set pRow = pCursor.NextRow
```

```
Do Until pRow Is Nothing
```

```
Count = Count + 1
```

```
xYAPStr = pRow.Value(labelPtFI) & pRow.Value(areaFI) & pRow.Value(perimeterFI)
If xYAPStr = lastXYAPStr Then
    pSelSet.Add pRow.OID
End If
lastXYAPStr = xYAPStr

If Count / 1000 = CInt(Count / 1000) Then
    Debug.Print Count & " " & pRow.Value(labelPtFI)
End If

If Count / 1000000 = CInt(Count / 1000000) Then
    Debug.Print Count
    Debug.Print "xy : " & pRow.Value(5)
    'UNCOMMENT the Exit Sub line below if you want to break the processing
    'tasks into batches of a million records
    'Exit Sub
End If

Set pRow = pCursor.NextRow
Loop

End Sub
```
